Transfer Learning for Latin and Chinese Characters with Deep Neural Networks

Dan Cireșan

June 10th, 2015

IDSIA, USI and SUPSI
Manno-Lugano, Switzerland
dan@idsia.ch

IJCNN 2012 paper: www.idsia.ch/~ciresan
DNN – state of the art for image classification

• Image classification (best result on 7 datasets, 2 won competitions):
  – MNIST, NIST SD 19, CASIA (IJCAI 2011, ICDAR 2011, CVPR 2012)
  – NORB jittered & cluttered (CVPR 2012)
  – CIFAR10 (CVPR 2012)
  – Traffic signs: GTSRB (IJCNN 2011)

• Segmentation:
  – neuronal structures in EM stacks (1th place at ISBI 2012 EM segmentation competition)
Motivation ...

• DNN have state of the art result on many image classification tasks

• Training Deep, Large NN on raw pixel images is computationally expensive (days to months/years on normal CPUs)

• Using GPUs helps, but it can still take hours or days

... and Method

• Try to reuse part of a trained net for learning another task
Deep, Large Neural Networks
Deep, Large Neural Networks

- Introduced by Fukushima (80) and refined by LeCun et al. (98), Riesenhuber et al. (99), Simard et al. (03), Behnke (03), Ciresan et al. (11)
Deep, Large Neural Networks

- Introduced by Fukushima (80) and refined by LeCun et al.(98), Riesenhuber et al.(99), Simard et al.(03), Behnke (03), Ciresan et al. (11)
- Combine convolutional, max pooling and fully connected layers
Deep, Large Neural Networks

• Introduced by Fukushima (80) and refined by LeCun et al.(98), Riesenhuber et al.(99), Simard et al.(03), Behnke (03), Ciresan et al. (11)

• Combine convolutional, max pooling and fully connected layers

• Hierarchical architecture designed for image recognition, loosely inspired from biology
Deep, Large Neural Networks

- Introduced by Fukushima (80) and refined by LeCun et al. (98), Riesenhuber et al. (99), Simard et al. (03), Behnke (03), Ciresan et al. (11)
- Combine convolutional, max pooling and fully connected layers
- Hierarchical architecture designed for image recognition, loosely inspired from biology
- Convolutional layers use 2D neighborhood information, and share weights
Deep, Large Neural Networks

- Introduced by Fukushima (80) and refined by LeCun et al. (98), Riesenhuber et al. (99), Simard et al. (03), Behnke (03), Ciresan et al. (11)
- Combine convolutional, max pooling and fully connected layers
- Hierarchical architecture designed for image recognition, loosely inspired from biology
- Convolutional layers use 2D neighborhood information, and share weights
- Fully supervised, with randomly initialized filters, trained by minimizing the misclassification error
Deep, Large Neural Networks

• Introduced by Fukushima (80) and refined by LeCun et al. (98), Riesenhuber et al. (99), Simard et al. (03), Behnke (03), Ciresan et al. (11)

• Combine convolutional, max pooling and fully connected layers

• Hierarchical architecture designed for image recognition, loosely inspired from biology

• Convolutional layers use 2D neighborhood information, and share weights

• Fully supervised, with randomly initialized filters, trained by minimizing the misclassification error

• Need GPU implementation to show their potential in a reasonable time

Transfer Learning for Latin and Chinese Characters with Deep Neural Networks; D. Ciresan et al.
Transfer Learning for Latin and Chinese Characters with Deep Neural Networks; D. Ciresan et al.
Training is computationally intensive: case study – Chinese OCR

<table>
<thead>
<tr>
<th>BIG NETWORK: 9HL-48x48-100C3-MP2-200C2-MP2-300C2-MP2-400C2-MP2-500N-3755</th>
<th>#weights</th>
<th>#connections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature extraction layers:</td>
<td>321’800</td>
<td>26.37 Million</td>
</tr>
<tr>
<td>Classification layer:</td>
<td>1’881’255</td>
<td>1.88 Million</td>
</tr>
</tbody>
</table>

1’121’749 training samples

CPU: **14 months** for 30 epochs of training
Graphics processing units (GPUs)

- 8 x GTX 480/580 1.5GB RAM
- >12 TFLOPS (theoretical speed) on 2 kW
- 40-80x speed-up compared with a single threaded CPU version of the CNN program (one day on GPU instead of two months on CPU)
- ~3000€ for one workstation with 4 GPUs

Or one week on GPU!
Experiments

• Methodology:
  – Fully train a net on one of the datasets (source dataset), then
  – Partially train the net on another dataset (destination dataset)

• Tasks:
  – Latin characters: from digits to uppercase letters
  – Learning uppercase letters from few samples per class
  – Chinese characters to uppercase Latin letters
  – Chinese characters: speeding up training
  – Uppercase letters to Chinese characters

Transfer Learning for Latin and Chinese Characters with Deep Neural Networks; D. Ciresan et al.
Datasets

- Handwritten Chinese characters
  - Subset of CASIA, 1000 classes out of 3755
  - 239121 characters for training and 59660 for testing
  - Scaled to 48x48 pixels

- Handwritten digits and Latin characters
  - NIST SD 19
  - Digits: 344307 for training and 58646 for testing
  - Uppercase letters: 69522 for training and 11941 for testing
  - Scaled to 29x29 pixels

Transfer Learning for Latin and Chinese Characters with Deep Neural Networks; D. Ciresan et al.
Enhance the training set with distorted samples

- Simulate expected variation in style:
  - Scaling factor: 0.85 - 1.15
  - Translation: maximum ±15%
  - Rotation: maximum ±15°
  - Elastic distortion

- The training set is distorted at the beginning of every epoch
Latin characters: from digits to uppercase letters

- Randomly initialized nets need to train all three convolutional layers.
- The feature extractors of nets pretrained on digits do not need retraining for the uppercase letters; retraining the fully connected layers suffices.
- Pretrained nets train faster.
• Pretrained nets train faster
Learning uppercase letters from few samples per class

- Nets are pretrained on DIGITS or just randomly initialized
- Retraining is done for all layers
- Without distortions, pretrained nets produce 36-70% of the randomly initialized nets errors
- Adding distortions alleviates the gap, but for less than 50 samples/class pretrained nets are far better
Chinese characters to uppercase Latin letters

- All characters are scaled to 48x48
- Pretrained nets are very good even if they are trained starting at the fourth convolutional layer
- The net trained on Chinese characters learns filter which can be fully reused for Latin letters
- Training only the output layer of a pretrained net is enough to obtain a low 3.35% error rate

Transfer Learning for Latin and Chinese Characters with Deep Neural Networks; D. Ciresan et al.
Chinese characters: speeding up training

- Pretraining on a fraction (1% or 10%) of the classes in the training set, then retraining on the entire data

- A net pretrained on Chinese 100 does not need retraining of the first two convolutional layers

- Pretrained nets reach low error much faster
• Pretraining on a fraction (1% or 10%) of the classes in the training set, then retraining on the entire data.

• A net pretrained on Chinese 100 does not need retraining of the first two convolutional layers.

• Pretrained nets reach low error much faster.

Transfer Learning for Latin and Chinese Characters with Deep Neural Networks; D. Ciresan et al.
Uppercase letters to Chinese characters

- Most challenging task:
  - Chinese characters are more complex than of Latin letters
  - Resizing uppercase letters from 29x29 to 48x48 changes the stroke thickness
- Pretraining on uppercase letters is almost as good as on Chinese 10
Conclusions

• In case of handwritten characters, DNN can be used for transfer learning between tasks

• Transfer learning from Latin letters to Chinese characters works as well as pretraining a net with 1% of the classes of the Chinese training task, despite the lower apparent complexity of Latin letters

• Advantages of transfer learning:
  – less training time is needed to obtain good results (5-25% of the normal training time)
  – much better results are obtained when only few labeled samples per class are available for the destination task

• Highly practical alternative to unsupervised training